



Multivariada I

- *Juliano van Melis – jvmelis@gmail.com*
- *Base: Profa. MSc. Edmila Montezani*
- *edmila@gmail.com*



Multivariada I

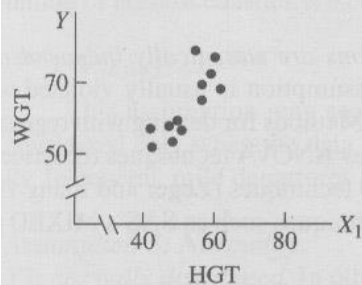
Introdução

- ◆ Pode ser vista como uma extensão da regressão simples
- ◆ Mais de uma variável independente é considerada.
- ◆ Lidar com mais de uma variável é mais difícil, pois:
 - ✓ É mais difícil escolher o melhor modelo, uma vez que diversas variáveis candidatas podem existir
 - ✓ É mais difícil visualizar a aparência do modelo ajustado, mais difícil a representação gráfica em mais de 3 dimensões
 - ✓ Às vezes, é difícil interpretar o modelo ajustado
 - ✓ Cálculos difíceis de serem executados sem auxílio de computador

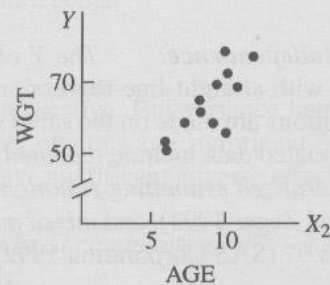
Exemplo: Supondo dados de peso, altura e idade de 12 crianças:

Criança	1	2	3	4	5	6	7	8	9	10	11	12
Peso (Y)	64	71	53	67	55	58	77	57	56	51	76	68
Altura (X ₁)	57	59	49	62	51	50	55	48	42	42	61	57
Idade (X ₂)	8	10	6	11	8	7	10	9	10	6	12	9

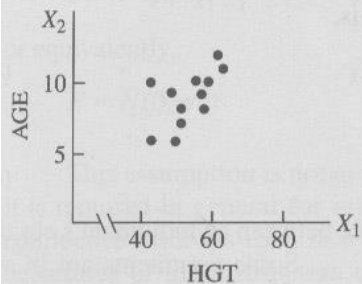
3 Separate scatter diagrams of WGT versus HGT, WGT versus AGE, and AGE versus HGT.



(a) WGT versus HGT ($r_{1Y} = 0.814$)



(b) WGT versus AGE ($r_{2Y} = 0.770$)



(c) AGE versus HGT ($r_{12} = 0.614$)

- ◆ A regressão múltipla pode ser usada para estudar o peso e sua variação em função da altura e idade das crianças.



Modelo

- ◆ O modelo de Regressão Linear Múltipla é representado pela equação:
-

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon$$

- ◆ As constantes: $\beta_0, \beta_1, \beta_2, \dots, \beta_k$, são os parâmetros populacionais.
- ◆ Os estimadores são representadas por: $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_k$



Exercícios



Regressão Linear

Agora que já aprendemos 'tudo' sobre ANOVA, vamos aprender um pouco sobre um outro tipo de modelo linear que é usado para analisar a relação entre variáveis contínuas: a *Regressão Linear*. A regressão linear é usada quando: (1) se quer saber se uma variável contínua está associada a outra variável contínua; (2) quando se quer medir a força da associação (r^2); ou (3) se quer a equação que descreve a relação para poder usá-la na predição de valores que não são conhecidos.

A regressão linear (simples ou múltipla) é feita com a função `lm()`, a exemplo do ANOVA - que também é um modelo linear. Relembrando, essa função requer uma fórmula. No caso da regressão linear simples a fórmula assume a forma $DV \sim IV$, que podemos ler como 'DV como função de IV' ou 'DV predita por IV', 'DV modelada por IV', etc. Lembre-se que DV (ou variável resposta) deve vir antes de '~' e IV ou variáveis explanatórias depois. É super simples, vamos tentar usando a base de dados 'cats' do pacote 'MASS', que contém informação sobre algumas características de gatos domésticos.

```
require(MASS)
```

```
## Loading required package: MASS
```

```
data(cats)
str(cats)
```

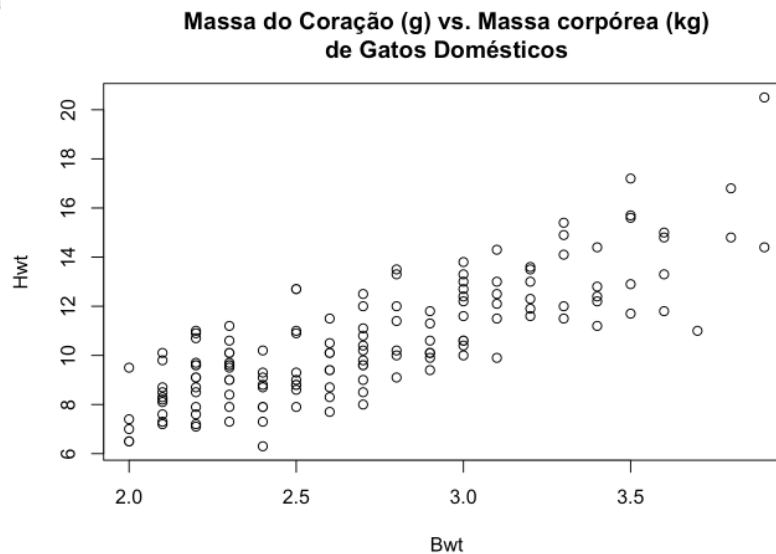
```
## 'data.frame':   144 obs. of  3 variables:
## $ Sex: Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
## $ Bwt: num  2 2 2 2.1 2.1 2.1 2.1 2.1 2.1 2.1 ...
## $ Hwt: num  7 7.4 9.5 7.2 7.3 7.6 8.1 8.2 8.3 8.5 ...
```

```
summary(cats)
```

```
## Sex      Bwt      Hwt
## F:47  Min.   :2.000  Min.   : 6.30
## M:97  1st Qu.:2.300  1st Qu.: 8.95
##      Median :2.700  Median :10.10
##      Mean   :2.724  Mean   :10.63
##      3rd Qu.:3.025  3rd Qu.:12.12
##      Max.   :3.900  Max.   :20.50
```

'Bwt' é a massa corpórea em kilogramas, 'Hwt' é a massa do coração em gramas, em machos e fêmeas ('Sex'). Vamos checar o comportamento dos dados usando um scatterplot.

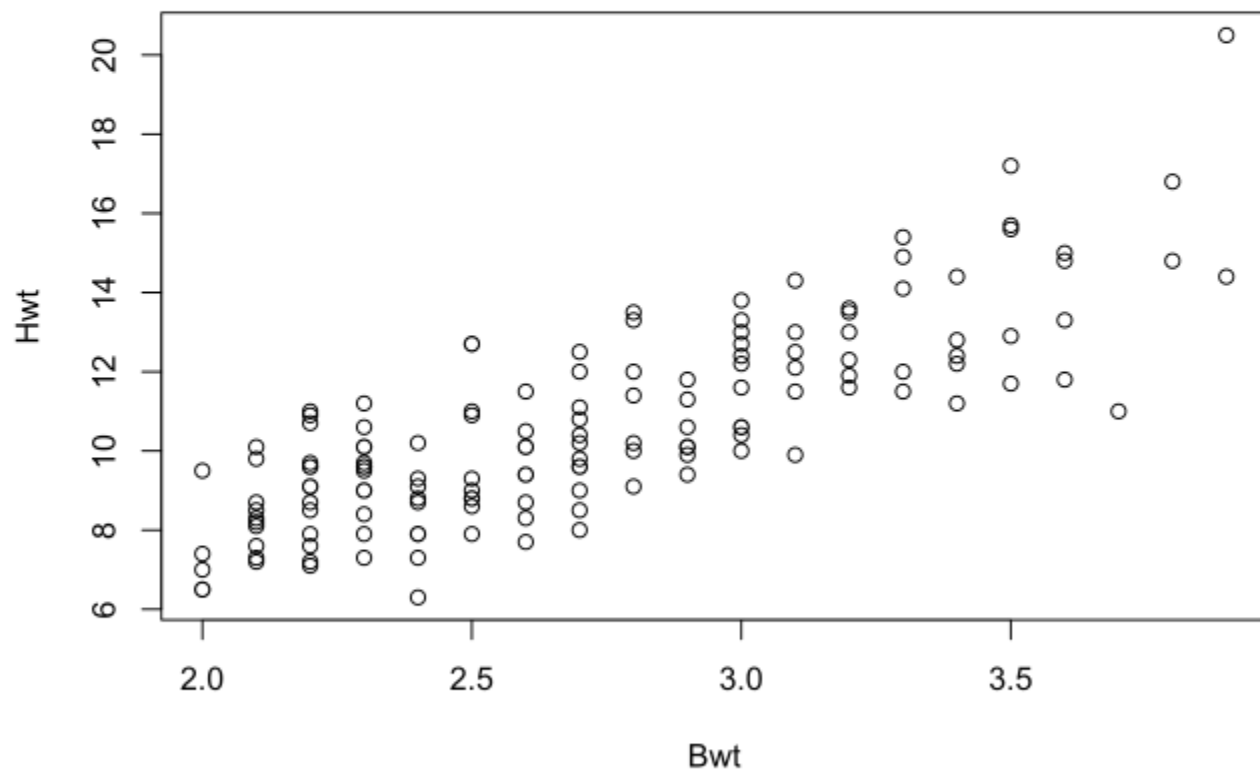
```
attach(cats)
plot(Bwt, Hwt)
title(main="Massa do Coração (g) vs. Massa corpórea (kg)\nde Gatos Domésticos")
```



A função `plot()` retorna um scatterplot sempre que alimentada com duas variáveis numéricas. A primeira variável listada é plotada no eixo horizontal. Também é possível usar a fórmula `DV ~ IV`, e o resultado é o mesmo.

```
plot(Hwt ~ Bwt, main="Massa do Coração (g) vs. Massa corpórea (kg)\nde Gatos Domésticos")
```

Massa do Coração (g) vs. Massa corpórea (kg) de Gatos Domésticos



O gráfico mostra uma relação que parece ser forte e razoavelmente linear entre a massa do coração e a massa corpórea. Vamos testar usando `lm()`



```
lm(Hwt ~ Bwt)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt)
##
## Coefficients:
## (Intercept)      Bwt
##      -0.3567      4.0341
```

Daí podemos tirar a equação da regressão: $Hwt = 4.0341 (Bwt) - 0.3567$. Mas para ter acesso a mais informações, é importante armazenar o output na forma de um objeto e extrair dele outras informações com outras funções.

```
fit <- lm(Hwt ~ Bwt)
fit
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt)
##
## Coefficients:
## (Intercept)      Bwt
##      -0.3567      4.0341
```



```
# reparem que essa função extrai muito mais informação
# incluindo o teste estatístico
summary(fit)
```

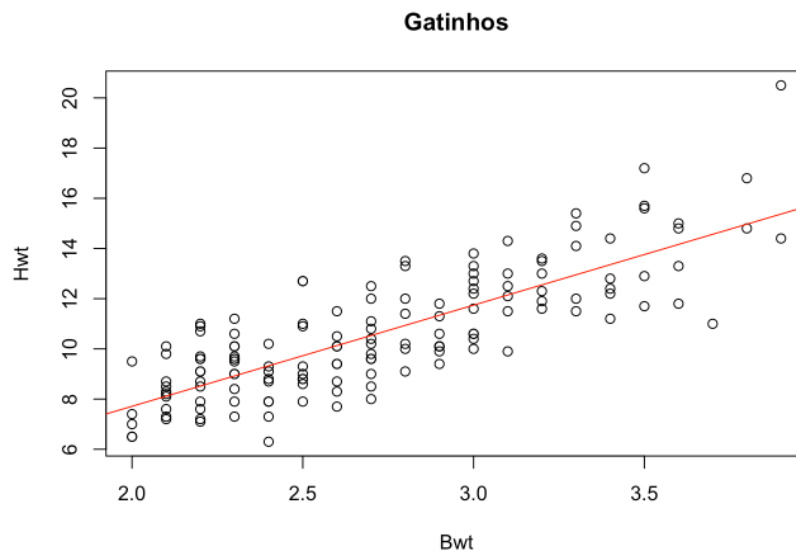
```
##
## Call:
## lm(formula = Hwt ~ Bwt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.5694 -0.9634 -0.0921  1.0426  5.1238
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3567     0.6923  -0.515   0.607
## Bwt           4.0341     0.2503  16.119 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.452 on 142 degrees of freedom
## Multiple R-squared:  0.6466, Adjusted R-squared:  0.6441
## F-statistic: 259.8 on 1 and 142 DF,  p-value: < 2.2e-16
```

```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: Hwt
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Bwt         1  548.09   548.09   259.83 < 2.2e-16 ***
## Residuals 142  299.53     2.11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

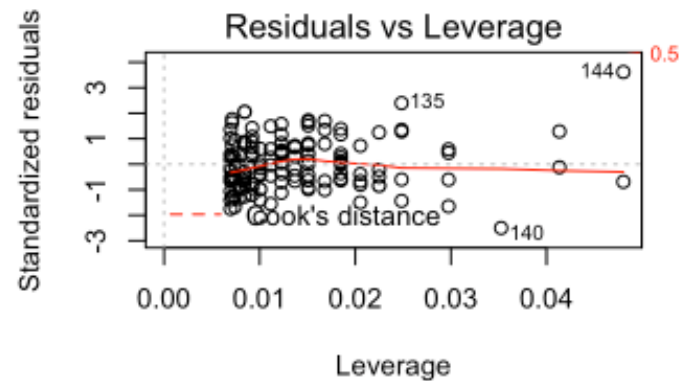
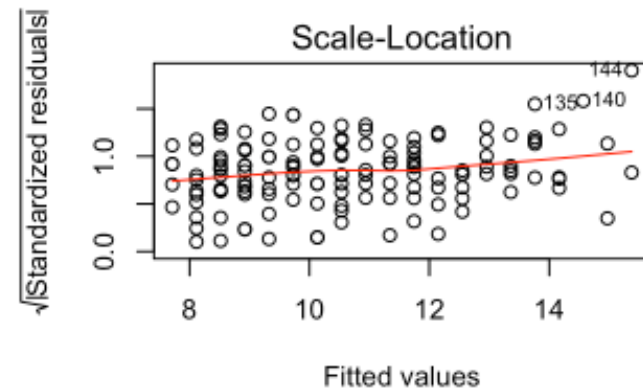
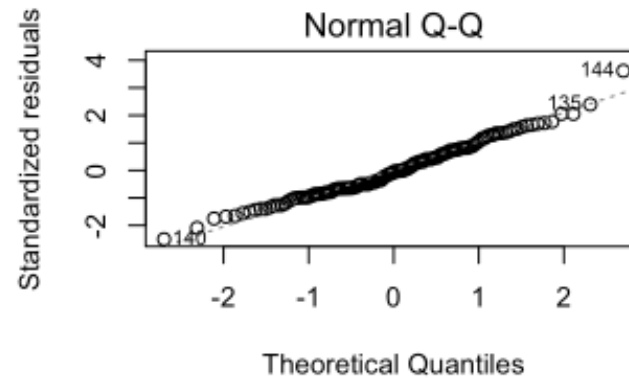
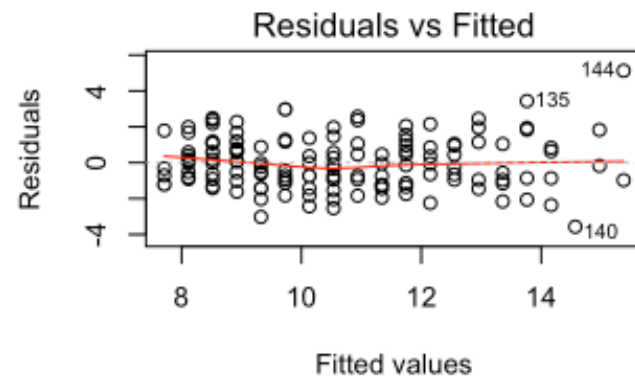
Nós podemos plotar a reta de regressão com o output do objeto para nos ajudar na interpretação. Para publicar nós vamos usar o ggplot2, que permite muito mais customização e fica bem mais bonito!


```
plot(Hwt ~ Bwt, main="Gatinhos")
abline(fit, col="red")
```



Há vários diagnósticos de regressão disponíveis, o mais simples dele pode ser feito usando a função `plot()`

```
par(mfrow=c(2,2))  
plot(fit)
```





O primeiro comando que usamos ajusta os parâmetros gráficos do dispositivo, particionando a janela gráfica em 4 partes (2 linhas e 2 colunas). Dessa forma, quatro gráficos podem ser plotados em apenas uma janela. A janela gráfica vai reter essa estrutura até que você a feche (no RSTUDIO seria o equivalente a usar o botão da vassoura na janela de plots), retornando então ao default. Outra opção é usar `par(mfrow=c(1,1))`.

O primeiro plot de diagnósticos é um *standard residual plot* mostrando os resíduos contra os valores ajustados. Pontos que apresentam uma tendência *outlier* são identificados. O que esperamos é que não apareça nenhuma tendência nos pontos deste plot, caso contrário o modelo linear pode não ser o mais adequado para estudar essa relação. O segundo plot é um *normal quantile plot*, que mostra se a distribuição dos resíduos segue ou não uma normal (esperamos que sim!). O último plot mostra *residuals vs. leverage*, cujos pontos identificados podem estar influenciando demais a tendência da regressão. Um deles é o caso 144.

```
cats[144, ]
```

```
##      Sex Bwt  Hwt  
## 144   M 3.9 20.5
```

```
fit$fitted[144]
```

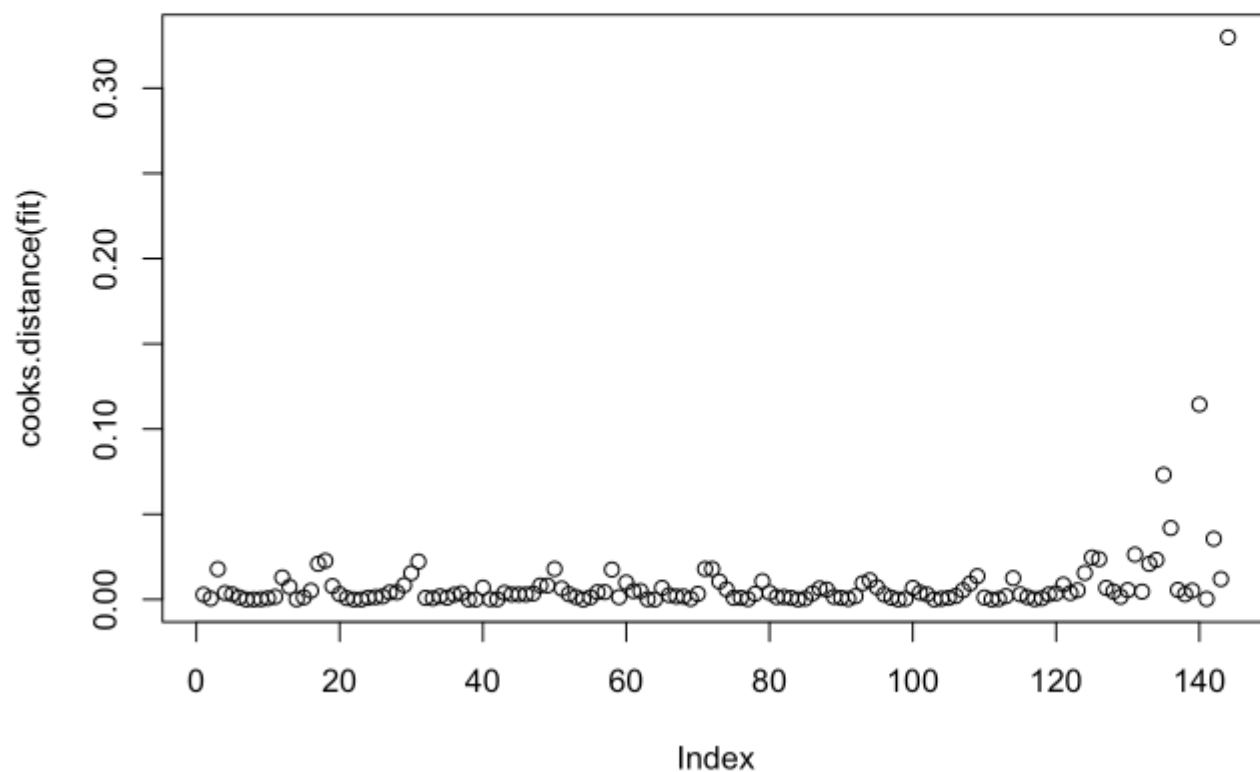
```
##      144  
## 15.37618
```

```
fit$residuals[144]
```

```
##      144  
## 5.123818
```

Ele era um gato gordo (3.9kg) e tinha um coração enorme (20.5g)! Os maiores de todo o dataset. O valor observado de 'Hwt' tem uma diferença (resíduo) de 5.1g em relação ao ajustado (15.4g). O erro padrão residual (vejo o output do modelo acima) foi 1.452. Convertendo para resíduo normalizado temos $5.124/1.452=3.53$, e isso é muito! Uma medida comum de influência seria *Cook's Distance*, vamos tentar.

```
par(mfrow=c(1,1))  
plot(cooks.distance(fit))
```



Qual o valor mais discrepante? E

agora o que fazer? Uma coisa que podemos fazer é ajustar um novo modelo sem o gato 144 e comparar os coeficientes.

```
fit.sem.144 <- lm(Hwt ~ Bwt, subset=(Hwt<20.5))
fit.sem.144
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt, subset = (Hwt < 20.5))
##
## Coefficients:
## (Intercept)      Bwt
##      0.118      3.846
```

Há também outras opções, como ajustar um modelo que é mais robusto tendo em vista que o dataset apresenta outliers. Existe uma função no pacote *MASS* chamada `rlm()` que é interessante para estes casos.

ANCOVA

A análise de covariância (ANCOVA) é usada para comparar duas ou mais retas de regressão, testando o efeito de um fator (variável categórica) em uma variável dependente (y) enquanto controla o efeito de uma co-variável contínua (x).

As retas são comparadas estudando a interação de uma variável categórica (por exemplo tratamentos, sexo) com a variável independente. Vamos tentar com os dados que já estávamos trabalhando. A pergunta que queremos testar é se machos e fêmeas são diferentes quanto ao tamanho do coração. Não podemos usar uma ANOVA pois o tamanho do coração depende do tamanho do corpo, como vimos no exemplo anterior.

```
fit2 <- aov(Hwt ~ Bwt * Sex)
summary(fit2)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Bwt           1  548.1   548.1 263.645 <2e-16 ***
## Sex           1    0.2     0.2  0.074 0.7853
## Bwt:Sex       1    8.3     8.3  4.008 0.0472 *
## Residuals    140 291.0     2.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Na ANCOVA, a primeira hipótese nula que testamos é a de que os coeficientes das retas de regressão (neste caso a reta das fêmeas e a dos machos) são iguais, ou seja as retas são paralelas. Isso é feito analisando a interação entre o fator e a covariável. Se a interação é significativamente diferente de zero, os efeitos da covariável na resposta dependem do nível do fator. Ou seja, as retas de regressão têm diferentes coeficientes de inclinação. Neste caso, vemos que o efeito do aumento no tamanho do corpo na massa do coração é distinto entre machos e fêmeas. Com a interação significativa, não podemos seguir com o teste da segunda hipótese nula da ANCOVA, a de que os interceptos das retas são iguais (os grupos não são diferentes para tratamento, sexo...). Temos que fazer análises separadas para cada sexo e podemos ver o resultado obtido de forma gráfica.

```
machos <- subset(cats, Sex=="M")
femeas <- cats[cats$Sex=="F",]

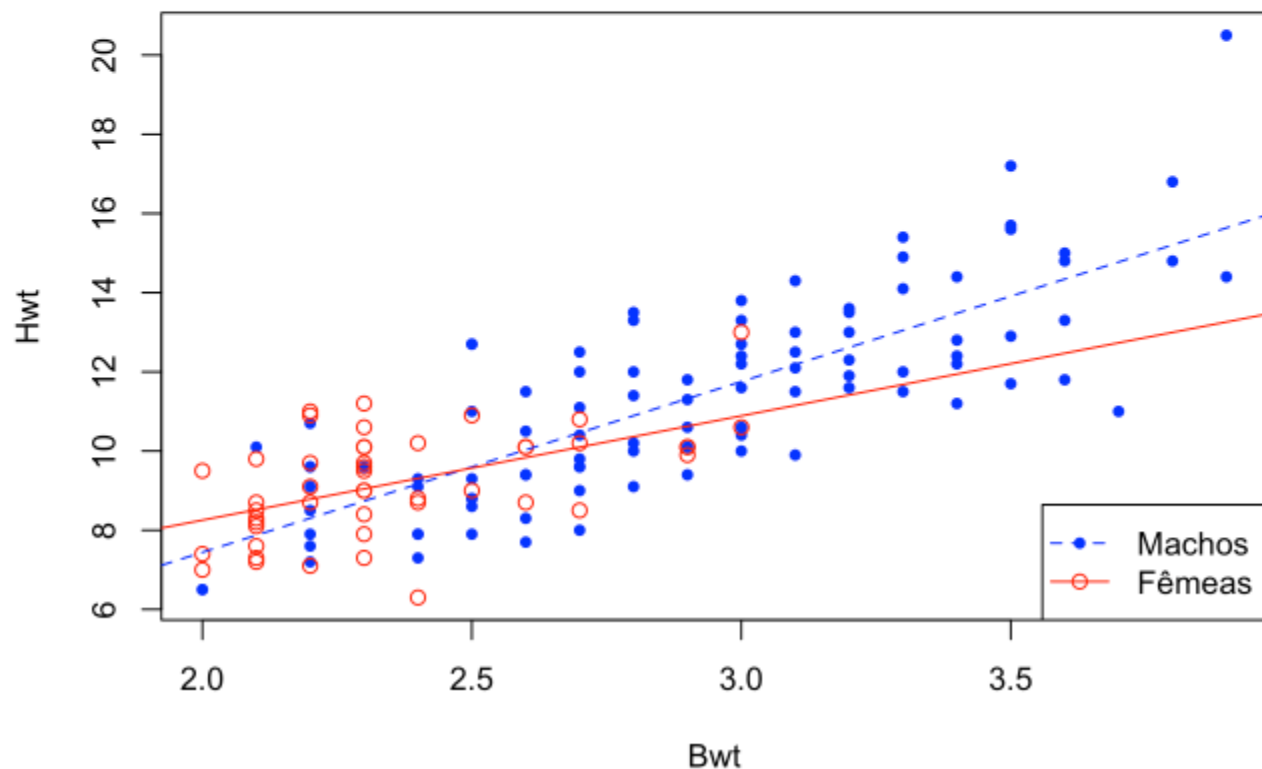
fitm <- lm(Hwt~Bwt, data=machos)
summary(fitm)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt, data = machos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.7728 -1.0478 -0.2976  0.9835  4.8646
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.1841     0.9983  -1.186   0.239
## Bwt           4.3127     0.3399  12.688 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.557 on 95 degrees of freedom
## Multiple R-squared:  0.6289, Adjusted R-squared:  0.625
## F-statistic: 161 on 1 and 95 DF, p-value: < 2.2e-16
```

```
fitf <- lm(Hwt~Bwt, data=femeas)
summary(fitf)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt, data = femeas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.00871 -0.68599 -0.04506  0.79583  2.21858
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9813     1.4855   2.007 0.050785 .
## Bwt           2.6364     0.6254   4.215 0.000119 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.162 on 45 degrees of freedom
## Multiple R-squared:  0.2831, Adjusted R-squared:  0.2671
## F-statistic: 17.77 on 1 and 45 DF,  p-value: 0.0001186
```

```
plot(Hwt~Bwt, type='n')
points(machos$Bwt,machos$Hwt, pch=20, col="blue")
points(femeas$Bwt,femeas$Hwt, pch=1, col='red')
abline(fitm, lty=2, col='blue')
abline(fitf, lty=1, col='red')
legend("bottomright", c("Machos","Fêmeas"), lty=c(2,1),
      pch=c(20,1), col=c('blue', 'red') )
```



Com um outro dataset, vamos testar se o número de sementes que uma determinada planta produz é alterado pela qualidade do solo, dada pelo tipo de fertilizante usado no experimento. Como sabemos que a produtividade dessa planta é uma função da chuva, usamos a quantidade de água oferecida a planta durante o experimento como covariável.

```
sementes <- read.csv('http://renatabrandt.github.io/EBC2015/data/sementes.csv')
sem.fit1 = lm(Sementes ~ Chuva*Fertilizante, data = sementes)
anova(sem.fit1)
```

```
## Analysis of Variance Table
##
## Response: Sementes
##
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
## Chuva	1	36068	36068	34423.6954	<2e-16 ***
## Fertilizante	2	24952	12476	11907.1841	<2e-16 ***
## Chuva:Fertilizante	2	1	1	0.7147	0.4895
## Residuals	1494	1565	1		

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

O resultado nos mostra que a quantidade de sementes produzidas depende da quantidade de água oferecida para a planta. Além disso, que essa relação não difere entre os tratamentos de fertilizantes. Podemos prosseguir com a análise para testar nossa hipótese principal retirando o termo de interação.

```
sem.fit2 = lm(Sementes ~ Chuva + Fertilizante, data = sementes)
anova(sem.fit2)
```

```
## Analysis of Variance Table
##
## Response: Sementes
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Chuva      1  36068   36068    34437 < 2.2e-16 ***
## Fertilizante 2  24952   12476    11912 < 2.2e-16 ***
## Residuals 1496   1567         1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(sem.fit1,sem.fit2)
```

```
## Analysis of Variance Table
##
## Model 1: Sementes ~ Chuva * Fertilizante
## Model 2: Sementes ~ Chuva + Fertilizante
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1    1494 1565.4
## 2    1496 1566.9 -2    -1.4978 0.7147 0.4895
```

O teste nos mostra que remover a interação não afeta o ajuste do modelo significativamente. Logo, ficamos com o modelo mais parsimonioso. Biologicamente então temos que a quantidade de sementes produzidas pelas nossas plantas aumenta com a quantidade de água fornecida para as plantas (proxy para chuva), e que os tratamentos com diferentes fertilizantes aumentam a produção de sementes. Mas queremos ver visualmente certo?

```
a <- subset(sementes, Fertilizante=="F1")
b <- subset(sementes, Fertilizante=="F2")
c <- subset(sementes, Fertilizante=="F3")
```

```
fita <- lm(Sementes ~ Chuva, data=a)
summary(fita)
```

```
##
## Call:
## lm(formula = Sementes ~ Chuva, data = a)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.7399 -0.6119  0.0064  0.7138  3.2488
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 104.11322    1.36555   76.24  <2e-16 ***
## Chuva        4.96353     0.04545  109.21  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9954 on 498 degrees of freedom
## Multiple R-squared:  0.9599, Adjusted R-squared:  0.9598
## F-statistic: 1.193e+04 on 1 and 498 DF,  p-value: < 2.2e-16
```

```
fitb <- lm(Sementes ~ Chuva, data=b)
summary(fitb)
```

```
##
## Call:
## lm(formula = Sementes ~ Chuva, data = b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.08953 -0.65306  0.01996  0.67215  2.91962
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  97.56039    1.38130   70.63  <2e-16 ***
## Chuva        5.01556    0.04597  109.10  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.007 on 498 degrees of freedom
## Multiple R-squared:  0.9598, Adjusted R-squared:  0.9598
## F-statistic: 1.19e+04 on 1 and 498 DF,  p-value: < 2.2e-16
```

```
fitc <- lm(Sementes ~ Chuva, data=c)
summary(fitc)
```

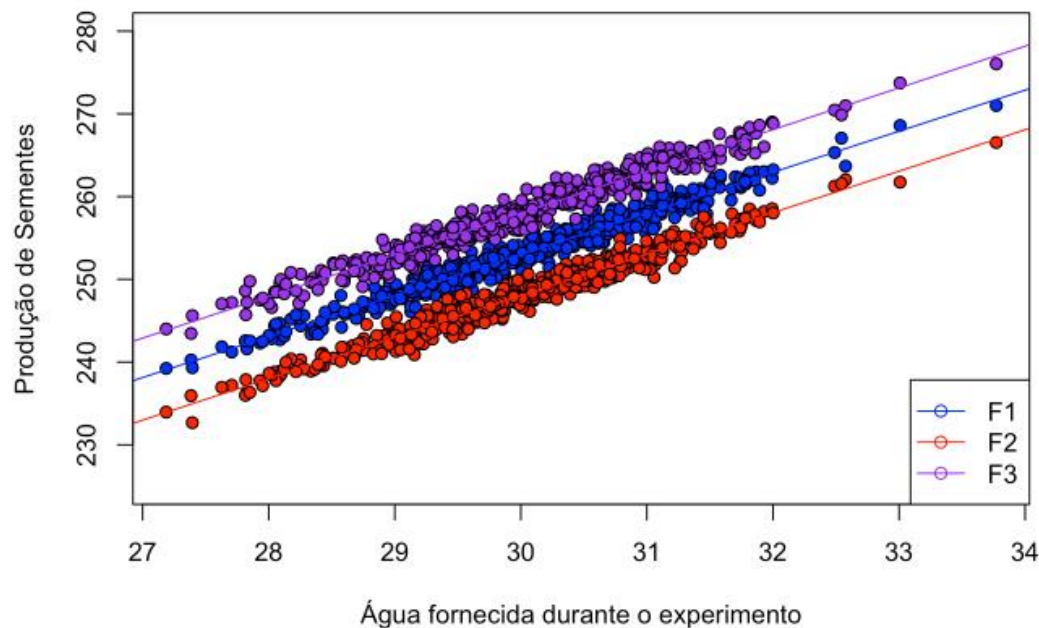
```
##
## Call:
## lm(formula = Sementes ~ Chuva, data = c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89939 -0.69463  0.01832  0.72268  2.84367
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 106.78492    1.46411   72.94  <2e-16 ***
## Chuva        5.04106     0.04873  103.45  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.067 on 498 degrees of freedom
## Multiple R-squared:  0.9555, Adjusted R-squared:  0.9554
## F-statistic: 1.07e+04 on 1 and 498 DF,  p-value: < 2.2e-16
```



```
attach(sementes)
plot(Sementes ~ Chuva, type='n', ylab = 'Produção de Sementes',
     xlab = 'Água fornecida durante o experimento', ylim = c(225,280))
points(a$Chuva,a$Sementes, pch=21, bg="blue")
points(b$Chuva,b$Sementes, pch=21, bg='red')
points(c$Chuva,c$Sementes, pch=21, bg="purple")

abline(fita, lty=1, col='blue')
abline(fitb, lty=1, col='red')
abline(fitc, lty=1, col='purple')

legend("bottomright", c("F1","F2", "F3"), lty= 1,
     pch=21, col=c('blue', 'red', 'purple') )
```



Exercício – Modelo Polinomial

Uma indústria está iniciando a produção de uma nova substância química. A meta é a produção da substância com um valor mínimo estabelecido para a variável *rendimento da reação química* (Y) a partir da qual a substância é produzida. As variáveis candidatas a preditoras são o *tempo de reação* (X_1) e a *temperatura do reator* (X_2).

Subir dados **quimica.csv**

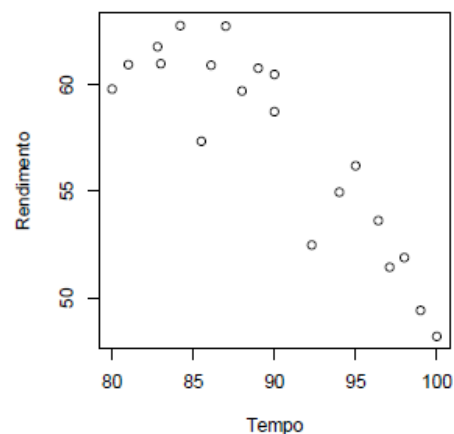
Tempo	Temperatura	Rendimento
81.0	173	60.95
85.5	168	57.35
83.0	185	60.99
94.0	188	54.96
98.0	183	51.89
97.1	175	51.44
82.8	173	61.79
89.0	183	60.78
92.3	168	52.48
80.0	175	59.8
90.0	188	58.74
95.0	179	56.2
90.0	181	60.49
84.2	177	62.78
88.0	171	59.71
87.0	182	62.75
99.0	184	49.41
96.4	187	53.63
100.0	180	48.19
86.1	172	60.92

```
> attach(quimica)
> quimica
      Tempo Temperatura Rendimento
1    81.0          173     60.95
2    85.5          168     57.35
3    83.0          185     60.99
.
.
```

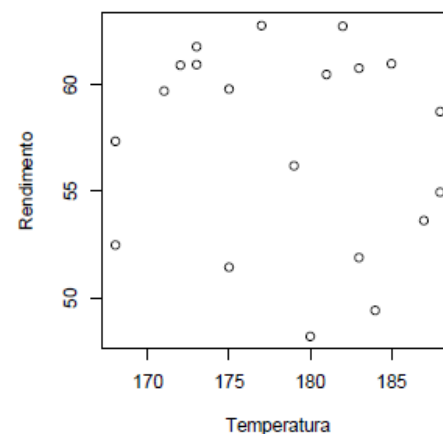
Veja a relação entre as variáveis explicativas e a variável resposta:

```
> par(mfrow=c(1,2))
> plot(Rendimento~Tempo)
> plot(Rendimento~Temperatura)
```

O diagrama de dispersão da Figura 11 (a) mostra uma relação curvilínea entre as variáveis Tempo e Rendimento. Já entre as variáveis Temperatura e Rendimento (Figura 11 (b)), não há uma relação bem definida.



(a) Tempo



(b) Temperatura

Figura 11: Diagramas de dispersão das Variáveis explicativas vs Variável resposta

Ajustaremos um modelo polinomial de grau 2 aos dados. Lembrando que as variáveis explicativas devem estar centradas em suas médias para o ajuste de um modelo polinomial, primeiramente realize tais centralizações pelos seguintes comandos:

```
> mean(Tempo)
[1] 89.92
> mean(Temperatura)
[1] 178.6
> Tempo1 = Tempo-mean(Tempo)
> Temperatura1 = Temperatura-mean(Temperatura)
```

O modelo ajustado abaixo, do *Rendimento* em *Tempo—Média* e *Temperatura—Média* é:

$$\begin{aligned}\hat{Y} = & 61,24 - 0,66(X_1 - 89,92) - 0,07(X_1 - 89,92)^2 \\ & + 0,12(X_2 - 178,60) - 0,04(X_2 - 178,60)^2 \\ & + 0,02(X_1 - 89,92)(X_2 - 178,60)\end{aligned}$$

com $R^2_{ajustado} = 0,99$. Para obter \hat{Y} em função de X_1 e X_2 , basta desenvolver a equação acima.

```
> ajuste = lm(Rendimento~Tempo1+I(Tempo1^2)+Temperatura1+
I(Temperatura1^2)+Tempo1*Temperatura1)
> summary(ajuste)
Call:
lm(formula = Rendimento ~ Tempo1 + I(Tempo1^2) + Temperatura1 +
    I(Temperatura1^2) + Tempo1 * Temperatura1)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.9491	-0.3643	0.1377	0.3071	0.5979

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	61.239297	0.284065	215.582	< 2e-16
Tempo1	-0.663385	0.021557	-30.773	2.94e-14
I(Tempo1^2)	-0.068866	0.004575	-15.053	4.86e-10
Temperatura1	0.115252	0.021065	5.471	8.24e-05
I(Temperatura1^2)	-0.042690	0.003956	-10.791	3.61e-08
Tempo1:Temperatura1	0.018875	0.004918	3.838	0.00181

Residual standard error: 0.5347 on 14 degrees of freedom
Multiple R-squared: 0.9902, Adjusted R-squared: 0.9867
F-statistic: 283 on 5 and 14 DF, p-value: 1.521e-13

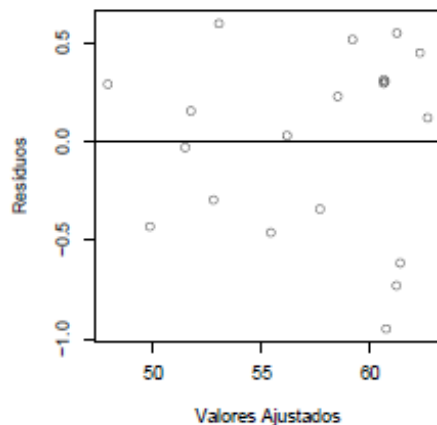
```
> shapiro.test(residuals(ajuste))
```

Shapiro-Wilk normality test

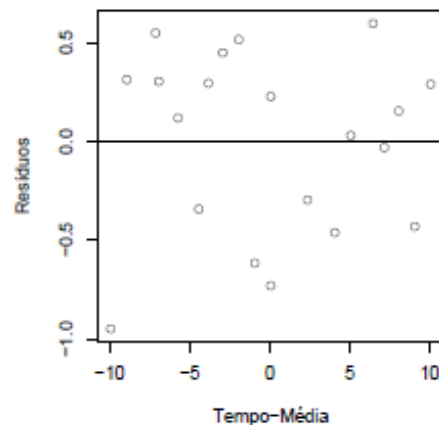
data: residuals(ajuste)

W = 0.9283, p-value = 0.1429

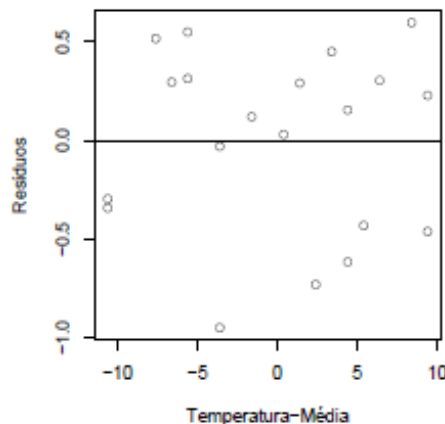
```
> windows()
> par(mfrow = c(2, 2))
> plot(fitted(ajuste), residuals(ajuste),xlab="Valores Ajustados",ylab="Resíduos")
> abline(h=0)
> plot(Tempo1, residuals(ajuste),xlab="Tempo-Média",ylab="Resíduos")
> abline(h=0)
> plot(Temperatura1, residuals(ajuste),xlab="Temperatura-Média",ylab="Resíduos")
> abline(h=0)
> qqnorm(residuals(ajuste), ylab="Resíduos",xlab="Quantis teóricos",main="")
> qqline(residuals(ajuste))
```



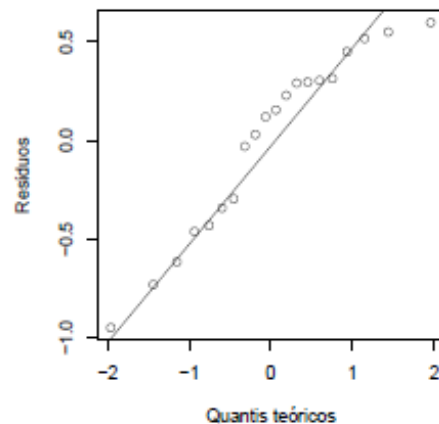
(a) Resíduos vs Valores ajustados



(b) Resíduos vs Tempo-Média



(c) Resíduos vs Temperatura-Média



(d) Gráfico de Probabilidade Normal dos Resíduos

A análise dos resíduos mostrada na Figura ao lado, indica que as condições de normalidade e ausência de correlação entre erros foram satisfeitas.

A normalidade ainda é confirmada pelo Teste de normalidade de *Shapiro-Wilk*, cujo *P-valor* é 0,1429.



Plotando Gráficos – Análise Multivariada

Plotando Gráficos – Análise Multivariada

Carregue o conjunto de dados “trees” contido no pacote **datasets**.

Os dados referem-se a 31 observações medidas de três variáveis: circunferência (em polegadas), altura (em pés) e volume (em pés cilíndricos) de madeira de cerejeiras negras.

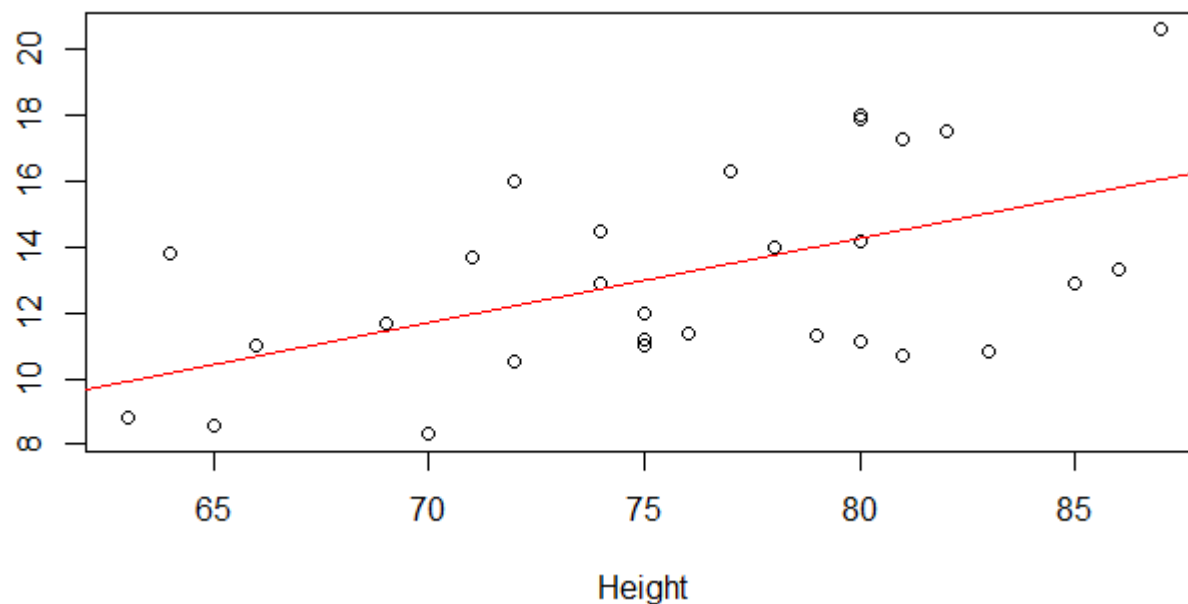
Temos:

```
require(datasets)
head(trees)
```

```
> head(trees)
   Girth Height Volume
1    8.3     70  10.3
2    8.6     65  10.3
3    8.8     63  10.2
4   10.5     72  16.4
5   10.7     81  18.8
6   10.8     83  19.7
```

Um gráfico de dispersão entre as duas primeiras variáveis pode ser obtido com o comando:

```
plot(Girth~Height, data=trees)  
abline(lm(Girth ~Height, data=trees), col="red")
```





Vamos agora visualizar o valor das correlações.

Isso pode ser feito com os argumentos `lower.panel` e `upper.panel` da seguinte forma:

#definindo uma função para desenhar retas de regressão:

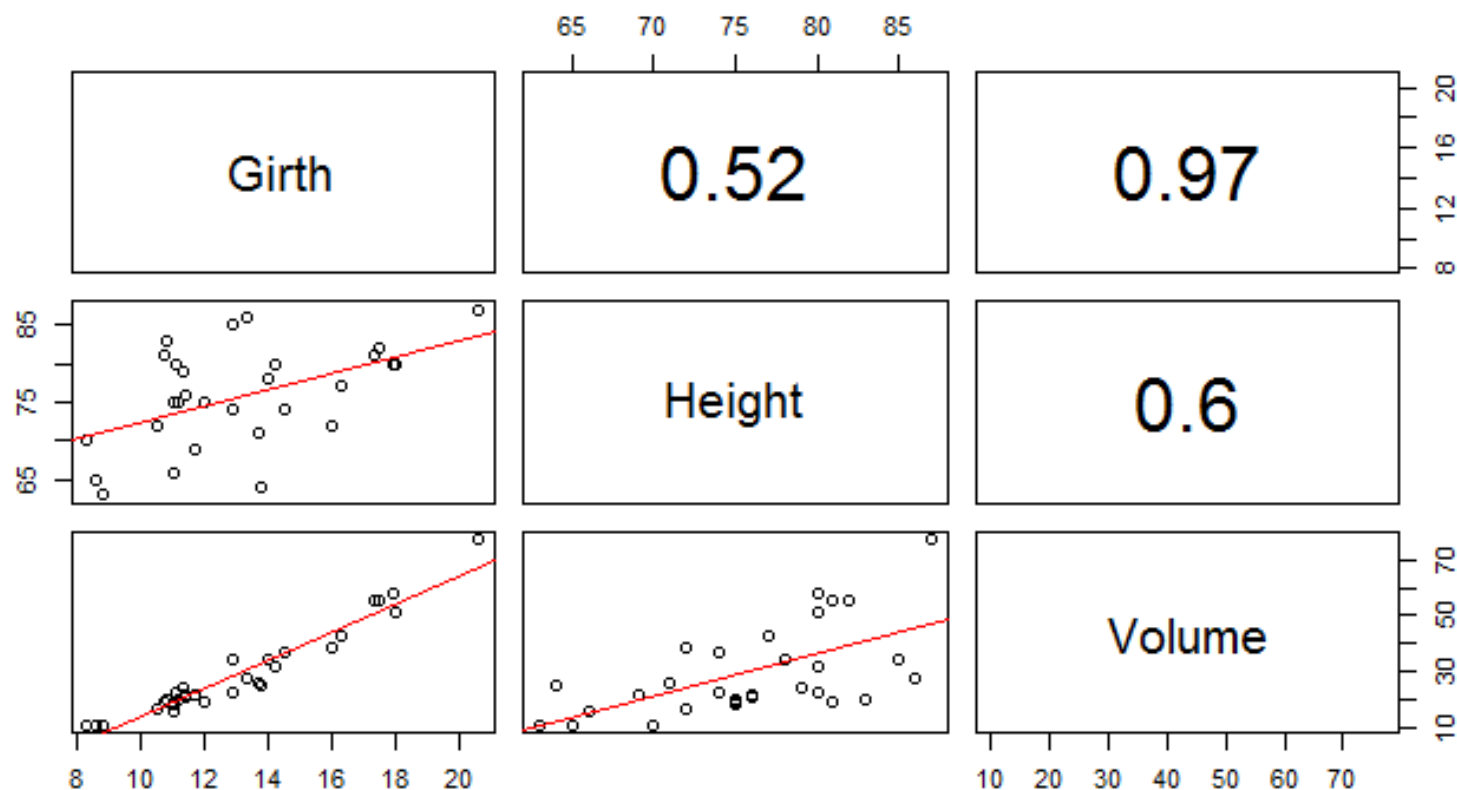
```
flines<- function(x,y){  
  points(x,y)  
  abline(lm(y~x), col="red")  
}
```

#definindo uma função para plotar as correlações:

```
fcor<- function(x,y){  
  par(usr=c(0,1,0,1))  
  txt<- as.character(round(cor(x,y),2))  
  text(0.5, 0.5, txt, cex=3)  
}
```

Vamos agora plotar o gráfico para essas correlações:

```
pairs(trees, lower.panel=flines, upper.panel = fcor)
```

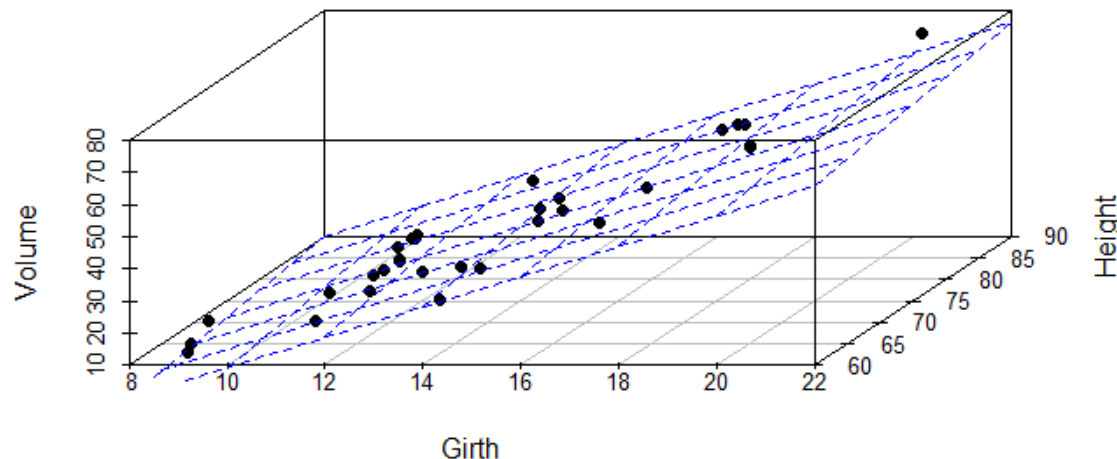


Visualização em 3D

Vamos usar o `scatterplot3d` para visualizar as três variáveis do dataset "trees" em um gráfico de dispersão e ainda inserir um plano de regressão.

Para isso, execute os comandos a seguir:

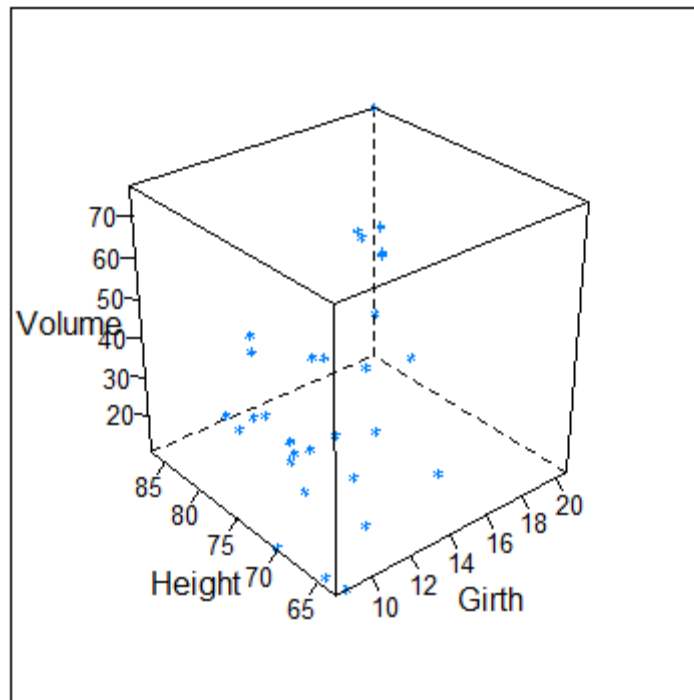
```
library(scatterplot3d)
attach(trees)
graph<- scatterplot3d(Volume ~Girth+Height, pch=16, angle=60)
fit<- lm(Volume~Girth +Height)
graph$plane3d(fit, col="blue")
```



Visualização em 3D

Se quiser usar o pacote `lattice`, a visualização é “semelhante”:

```
library(lattice)
cloud(Volume ~ Girth*Height, data=trees,
      scales=list(arrows=FALSE))
```



Visualização em 3D

Se quiser usar o pacote `rgl`, o gráfico fica mais “iterativo” – visto de frente:

```
library(rgl)
plot3d(trees)
```

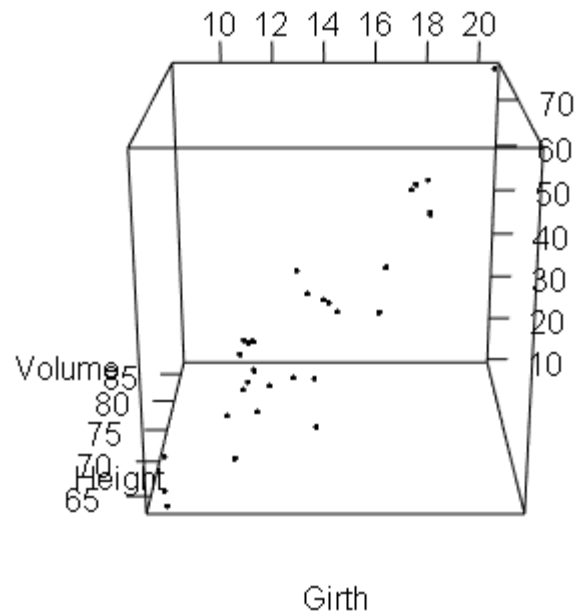




Gráfico de Estrelas

Outra forma de observações multivariadas é por meio do gráfico de estrelas.

Com o pacote `graphics`, utilizamos a função `stars()`, que permite desenhar diagramas de estrelas.

Por exemplo: Utilizando as cinco primeiras linhas (marcas de carros) e as duas primeiras colunas de `"mtcars"` (duas primeiras variáveis), utilizamos os comandos:

(dados `mtcars` são dados extraídos da Motor Trend US Magazine, em 1974, comparando o consumo de gasolina e 10 outros aspectos sobre design e performance para 32 marcas de automóveis – veja `help` do R `"mtcars"`)

```
require(graphics)
head(mtcars)
stars(mtcars[1:5, 1:2], nrow=2, key.loc=c(6.8, 1.8), draw.segments = TRUE,
col.segments = 1:2)
```

Variáveis:

1. Milhas por Galão
2. Número de cilindros

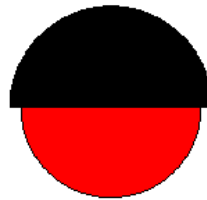
Gráfico de Estrelas

```
> head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1



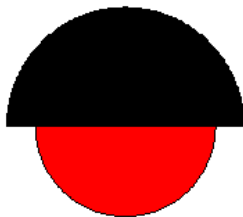
Mazda RX4



Mazda RX4 Wag



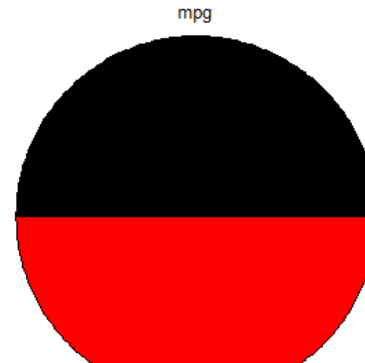
Datsun 710



Hornet 4 Drive



Hornet Sportabout



mpg

Faces de Chernoff

A representação multivariada de observações ou indivíduos consiste em representar os valores das diversas variáveis por meio de símbolos ou formas geométricas.

Nesse sentido, gráficos como os de Chernoff apelam para o VISUAL 😊

Considere os dados disponíveis no pacote dataset “mtcars”.

Há 32 observações (linhas) e 11 colunas (variáveis).

As últimas 6 linhas são:

```
tail(mtcars)
```

```
> tail(mtcars)
      mpg  cyl  disp  hp drat   wt  qsec vs  am  gear  carb
Porsche 914-2 26.0   4 120.3  91 4.43 2.140 16.7  0   1     5     2
Lotus Europa 30.4   4  95.1 113 3.77 1.513 16.9  1   1     5     2
Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0   1     5     4
Ferrari Dino 19.7   6 145.0 175 3.62 2.770 15.5  0   1     5     6
Maserati Bora 15.0   8 301.0 335 3.54 3.570 14.6  0   1     5     8
Volvo 142E 21.4   4 121.0 109 4.11 2.780 18.6  1   1     4     2
```

Faces de Chernoff

Baixe agora o pacote TeachingDemos:

```
install.packages("TeachingDemos")  
library(TeachingDemos)
```

E execute a seguinte função:

```
faces(tail(mtcars))
```

Porsche 914-2



Lotus Europa



Ford Pantera L



Ferrari Dino



Maserati Bora



Volvo 142E



Faces de Chernoff

Se quiser saber a classificação de todos os carros da lista:

`faces (mtcars)`

Mazda RX4



Mazda RX4 Wag



Datsun 710



Hornet 4 Drive



Hornet Sportabout



Valiant



Duster 360



Merc 240D



Merc 230



Merc 280



Merc 280C



Merc 450SE



Merc 450SL



Merc 450SLC



Cadillac Fleetwood



Lincoln Continental



Chrysler Imperial



Fiat 128



Honda Civic



Toyota Corolla



Toyota Corona



Dodge Challenger



AMC Javelin



Camaro Z28



Pontiac Firebird



Fiat X1-9



Porsche 914-2



Lotus Europa



Ford Pantera L



Ferrari Dino



Maserati Bora



Volvo 142E





Elipses de Correlação

Elipses de correlação podem ser de grande valia quando se deseja representar graficamente uma matriz de correlações, isto é, verificar a força e a direção das associações para um número considerável de variáveis, ou mesmo quando se deseja observar correlações em blocos ou grupos de variáveis.

Vamos nos basear no dataset `"mtcars"`.

Premissas:

→ Correlações próximas de 0 se aproximam do branco e aquelas próximas de 1 ou -1 são representadas por elipses mais escuras. (Isso pode ser feito com o auxílio da função `color.scale()`, do pacote `plotrix`.

```
install.packages("ellipse")  
library(ellipse)
```

```
install.packages("plotrix")  
library(plotrix)
```

Elipses de Correlação

```
#matrix de correlação (11 x 11)  
mcor<- cor(mtcars)
```

```
#matrix de cores  
mcores<- color.scale(1-abs(mcor))
```

```
#elipses  
plotcorr(mcor, col=mcores)
```

```
#delimitando cores  
abline(h=4.5, v=6.5, col="red")
```

